# Penerapan *Deep Learning* pada Sistem Deteksi Kemiripan Karya Citra Digital untuk Seleksi Pendaftaran Hak Cipta Berbasis *Artificial Intelligence*

1st Hafizh Ghiyats Ash Shiddiq
Fakultas Teknik Elektro
Universitas Telkom
Bandung, Indonesia
hafizhghiyatsash@ student.telkomuniversity.ac.id

2nd Iwan Iwut Tritoasmoro
Fakultas Teknik Elektro
Universitas Telkom
Bandung, Indonesia
iwaniwut@telkomuniversity.ac.id

3nd Eng Danu Dwi Sanjoyo
Fakultas Teknik Elektro
Universitas Telkom
Bandung, Indonesia
danudwj@telkomuniversity.ac.id

***Abstrak* — Peningkatan jumlah pendaftaran hak cipta karya citra digital menimbulkan tantangan dalam proses pemeriksaan kemiripan karya secara manual karena membutuhkan waktu yang lama dan rentan terhadap subjektivitas penilaian. Penelitian ini bertujuan mengembangkan sistem deteksi kemiripan karya citra digital untuk mendukung proses seleksi pendaftaran hak cipta berbasis *Artificial Intelligence*. Sistem dikembangkan dalam bentuk aplikasi berbasis web dengan memanfaatkan model *Contrastive Language-Image Pre-training* (CLIP) dan *Convolutional Neural Network* (CNN) untuk melakukan ekstraksi fitur citra dan perhitungan tingkat kemiripan menggunakan metode *cosine similarity*. Sistem menggunakan MongoDB sebagai basis data metadata dan Milvus sebagai basis data vektor untuk menyimpan *embedding* citra. Hasil pengujian menunjukkan bahwa sistem mampu mendeteksi citra identik serta citra yang mengalami perubahan *resize*, *crop*, dan rotasi dengan baik. Berdasarkan evaluasi menggunakan data internal Bandung Techno Park, sistem memperoleh nilai *accuracy* sebesar 95,94%, *precision* sebesar 99,33%, *recall* sebesar 92,50%, dan *F1-score* sebesar 95,79%. Selain itu, hasil pengujian *User Experience* menunjukkan bahwa sistem memperoleh kategori sangat baik dengan nilai rata-rata 94,798% pada seluruh aspek pengujian.**

***Kata kunci* — *Artificial Intelligence*, *Convolutional Neural Network* (CNN), *Contrastive Language-Image Pre-training* (CLIP), *cosine similarity*, deteksi kemiripan citra.**

## I. PENDAHULUAN

Citra digital didefinisikan sebagai representasi visual yang dikonversi menjadi data numerik sehingga memungkinkan untuk diproses dan dianalisis secara komputasional. Proses konversi ini melibatkan mesin di mana representasi tersebut diperoleh melalui pendekatan berdasarkan sampling dan kuantisasi [1]. Karya digital sendiri mencakup berbagai bentuk hasil ciptaan seperti ilustrasi, desain grafis, dan karya seni berbasis digital lainnya. Dengan perkembangan teknologi informasi membuat proses pembuatan, pengunggahan, dan penyebaran karya digital menjadi semakin mudah. Namun, kemudahan ini menimbulkan permasalahan terutama terkait perlindungan hak cipta dan keaslian karya [2]. Banyak kasus di mana karya digital digunakan kembali atau dimodifikasi tanpa izin sehingga menimbulkan kerugian bagi pencipta aslinya dan menurunkan nilai orisinalitas karya tersebut [3].

Berdasarkan hasil *interview* dengan pihak Bandung Techno Park, proses pendaftaran hak cipta untuk memastikan keaslian karya, saat ini masih dilakukan secara manual dengan menggunakan Google Lens. Pemeriksaan ini akan memakan waktu lama, terutama jika jumlah karya yang didaftarkan dalam jumlah yang banyak dan memiliki kemiripan satu sama lain [4]. Kondisi tersebut menyebabkan adanya keterlambatan dalam perlindungan hukum dan ketidakakuratan akibat keterbatasan manusia dalam mengidentifikasi kemiripan karya secara mendetail.

Permasalahan menjadi semakin kompleks karena kemiripan karya digital tidak terbatas pada duplikasi yang identik. Banyak karya yang jika dilihat oleh mata manusia secara visual tampak berbeda, akan tetapi mengandung elemen yang serupa karena hasil dari modifikasi kecil seperti rotasi dan *cropping* [5]. Kondisi tersebut akan membuat proses deteksi manual sulit dilakukan. Diperlukan suatu sistem yang mampu mengenali kesamaan tidak hanya dari sisi visual tetapi juga kemiripan semantik yaitu kesamaan makna dari citra tersebut. Sistem tradisional yang hanya membandingkan pola *pixel* dinilai kurang efektif untuk mendeteksi kemiripan seperti ini.

Permasalahan lain yang perlu diperhatikan terkait penentuan nilai ambang kemiripan atau *threshold* secara tepat agar sistem tidak terlalu sensitif atau menghasilkan banyak *false positive* maupun terlalu longgar atau menghasilkan banyak *false negative* [6]. Selain itu, proses verifikasi keberadaan karya citra digital di internet sering kali belum terintegrasi secara optimal, sehingga menyulitkan pemeriksaan apakah karya tersebut orisinal atau sudah pernah dipublikasikan sebelumnya.

Berdasarkan berbagai permasalahan tersebut, diperlukan suatu sistem deteksi kemiripan karya citra digital yang mampu melakukan proses analisis secara otomatis dan lebih

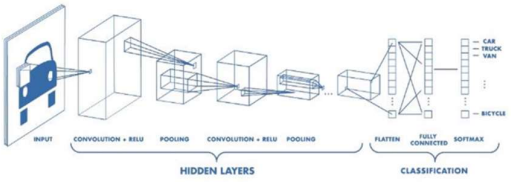
akurat. Sistem ini dirancang dengan memanfaatkan teknologi *Artificial Intelligence* (AI) untuk mengidentifikasi kemiripan visual maupun semantik pada karya digital, termasuk karya yang telah mengalami modifikasi tertentu. Selain itu, sistem juga dapat digunakan untuk melakukan pencarian karya serupa di internet guna membantu proses verifikasi keaslian dan mengetahui apakah karya tersebut pernah dipublikasikan sebelumnya. Dengan adanya sistem ini, proses pemeriksaan hak cipta diharapkan dapat dilakukan dengan lebih cepat, efektif, dan efisien dibandingkan metode manual yang digunakan saat ini.

## II. KAJIAN TEORI

### A. *Convolutional Neural Network*

*Convolutional Neural Network* atau CNN merupakan salah satu metode *deep learning* yang dirancang untuk mengolah data berupa citra atau gambar. CNN banyak diterapkan dalam bidang *computer vision* karena memiliki kemampuan dalam mengenali pola, bentuk, tekstur, serta objek pada citra secara otomatis. Berbeda dengan metode konvensional yang memerlukan proses ekstraksi fitur secara manual, CNN mampu mempelajari dan mengekstraksi fitur penting langsung dari data citra yang digunakan [7].

CNN terdiri dari beberapa lapisan utama, yaitu *convolution layer*, *activation layer*, *pooling layer*, dan *fully connected layer*. *Convolution layer* berperan dalam mengekstraksi fitur visual dari citra melalui penggunaan *kernel* atau *filter*. *Activation layer*, seperti ReLU, digunakan untuk membantu jaringan mempelajari pola yang lebih kompleks. Selanjutnya, *pooling layer* berfungsi mengurangi ukuran *feature map* agar proses komputasi menjadi lebih efisien, sedangkan *fully connected layer* digunakan untuk mengolah fitur yang telah diperoleh menjadi hasil akhir, seperti klasifikasi objek atau representasi fitur citra [8]. Berdasarkan uraian mengenai lapisan-lapisan penyusun CNN, arsitektur dasar CNN secara keseluruhan ditunjukkan pada Gambar 1.



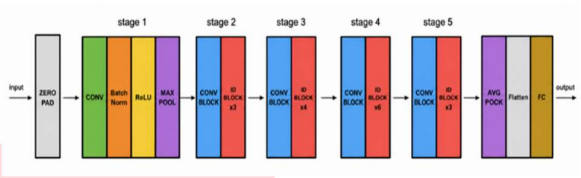

GAMBAR 1
Arsitektur dasar *Convolutional Neural Network* (CNN)

Berdasarkan Gambar 1, CNN dapat dipahami sebagai model yang mengubah citra masukan menjadi representasi fitur melalui beberapa tahapan pemrosesan. Representasi fitur tersebut menjadi bagian penting dalam berbagai aplikasi pengolahan citra karena dapat digunakan untuk mengenali pola visual pada gambar [9].

### B. *Residual Network*-50 (ResNet-50)

ResNet-50 (*Residual Network*-50) merupakan salah satu varian dari arsitektur *Residual Network* (ResNet) yang dikembangkan untuk mengatasi permasalahan *vanishing gradient* pada jaringan *Convolutional Neural Network* (CNN) yang memiliki banyak lapisan. Berbeda dengan arsitektur CNN konvensional yang mengalami penurunan performa ketika jumlah lapisan semakin dalam, ResNet-50 memperkenalkan konsep *residual learning* melalui mekanisme *shortcut connection* (*skip connection*), sehingga proses pembelajaran menjadi lebih stabil dan mampu mempertahankan informasi penting dari lapisan sebelumnya. Dengan kemampuan tersebut, ResNet-50 banyak dimanfaatkan pada berbagai tugas *computer vision*, seperti klasifikasi citra, deteksi objek, segmentasi citra, hingga image retrieval [10] [11].

Susunan arsitektur ResNet-50 secara keseluruhan dapat dilihat pada Gambar 2.




GAMBAR 2
Arsitektur ResNet-50

Berdasarkan Gambar 2, proses ekstraksi fitur pada ResNet-50 dimulai dari tahap *Zero Padding* untuk mempertahankan dimensi citra, kemudian dilanjutkan dengan *Convolution Layer*, *Batch Normalization*, ReLU, dan *Max Pooling*. Selanjutnya, citra diproses melalui empat kelompok *residual block* yang terdiri atas *Convolution Block* dan *Identity Block* untuk menghasilkan representasi fitur secara bertingkat. Setelah seluruh fitur diekstraksi, dilakukan *Global Average Pooling* untuk merangkum informasi spasial sebelum diteruskan menuju *Fully Connected Layer* sebagai tahap akhir proses klasifikasi. Arsitektur tersebut memungkinkan ResNet-50 menghasilkan representasi fitur yang kaya dengan proses pelatihan yang lebih stabil dibandingkan arsitektur CNN konvensional [10] [11].

Konsep utama pada ResNet-50 adalah *residual learning*, yaitu pendekatan yang tidak secara langsung mempelajari fungsi pemetaan H(x), melainkan mempelajari fungsi residual F(x). Hubungan tersebut dinyatakan pada Persamaan

$$F(x) = H(x) - x \quad (1)$$

sehingga keluaran dari suatu *residual block* dapat dituliskan sebagai

$$y = F(x) + x \quad (2)$$

dengan merupakan fitur masukan (*input feature*), F(x)merupakan fungsi residual yang dipelajari oleh beberapa lapisan konvolusi, sedangkan merupakan fitur keluaran (*output feature*). Penambahan shortcut *connection* memungkinkan informasi dari lapisan sebelumnya diteruskan secara langsung ke lapisan berikutnya sehingga aliran gradien tetap terjaga selama proses pelatihan. Mekanisme tersebut menjadikan ResNet-50 mampu dilatih pada jaringan yang lebih dalam dengan risiko vanishing gradient yang lebih kecil dibandingkan CNN konvensional [11].

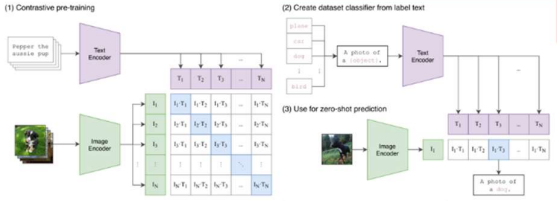
### C. *Contrastive Laguage Image Pre-training* (CLIP)

*Contrastive Language–Image Pre-training* (CLIP) merupakan model *vision-language* yang dikembangkan oleh OpenAI untuk mempelajari hubungan antara citra (*image*) dan bahasa alami (*natural language*) melalui proses *pre-training* menggunakan pasangan citra dan teks (*image-text*

*pairs*) dalam jumlah besar. Berbeda dengan model klasifikasi citra konvensional yang hanya mempelajari sejumlah kelas tertentu, CLIP memanfaatkan informasi visual dan deskripsi teks secara bersamaan sehingga mampu memahami hubungan semantik antara keduanya. Pendekatan tersebut memungkinkan model mempelajari representasi visual yang lebih umum dan tidak bergantung pada label kelas yang telah ditentukan sebelumnya [12].

Proses *pre-training* pada CLIP dilakukan menggunakan sekitar 400 juta pasangan citra dan teks yang dikumpulkan dari berbagai sumber di internet dan dikenal sebagai *WebImageText* (WIT). Melalui proses tersebut, model belajar menghubungkan karakteristik visual suatu citra dengan deskripsi teks yang sesuai sehingga mampu memahami berbagai konsep visual secara lebih luas. Pemanfaatan data dalam skala besar menjadikan CLIP memiliki kemampuan generalisasi yang lebih baik dibandingkan model yang hanya dilatih menggunakan dataset berlabel dengan jumlah kelas yang terbatas.[12].

Adapun arsitektur CLIP secara keseluruhan ditunjukkan pada Gambar 3.




GAMBAR 3
Arsitektur *Contrastive Language-Image Pre-training* (CLIP)

Berdasarkan Gambar 3, citra diproses oleh *image encoder* untuk menghasilkan *image embedding*, sedangkan deskripsi teks diproses oleh *text encoder* untuk menghasilkan *text embedding*. Kedua representasi tersebut kemudian diproyeksikan ke dalam *shared embedding space* sehingga hubungan semantik antara citra dan teks dapat direpresentasikan dalam ruang vektor yang sama. Representasi bersama tersebut menjadi dasar bagi CLIP untuk menghubungkan informasi visual dan bahasa alami, sehingga dapat dimanfaatkan pada berbagai tugas seperti *image retrieval*, *image classification*, maupun *zero-shot learning*

Untuk mengoptimalkan proses pembelajaran, CLIP menggunakan fungsi kerugian I*nfoNCE Loss* (*Normalized Temperature-scaled Cross Entropy Loss*). Fungsi ini bertujuan memaksimalkan nilai kemiripan antara pasangan positif sekaligus meminimalkan kemiripan terhadap seluruh pasangan negatif di dalam satu *batch*. Persamaan *InfoNCE Loss* ditunjukkan pada Persamaan (2.3).

$$L_1 = -\log \frac{\exp\left(sim(I_i, T_i)/\tau\right)}{\sum_{j=1}^{N} \left(sim(I_i, T_i)/\tau\right)} \qquad (3)$$

*dengan:*

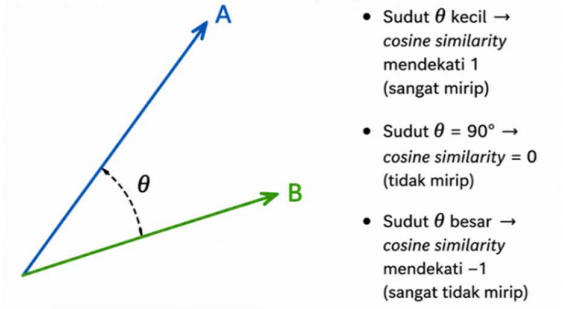
- $I_i$ = *image embedding ke-i*
- $T_i$ = *text embedding* yang sesuai
- $sim(I_i, T_j)$ = nilai *Cosine* Similarity antara citra dan teks
- $\tau$ = *parameter temperature*
- $N$ = jumlah pasangan citra dan teks dalam satu *batch*

Berdasarkan persamaan tersebut, nilai *loss* akan semakin kecil apabila kemiripan antara pasangan citra dan teks yang sesuai semakin tinggi dibandingkan dengan pasangan lainnya. Sebaliknya, apabila model masih memberikan nilai kemiripan yang tinggi pada pasangan yang tidak sesuai, maka nilai *loss* akan meningkat sehingga parameter model akan diperbarui selama proses pelatihan. [12] [13].

*D. Cosine Similarity*

*Cosine Similarity* merupakan metode yang digunakan untuk mengukur tingkat kemiripan antara dua vektor berdasarkan sudut yang terbentuk di antara keduanya. Metode ini banyak digunakan pada bidang *machine learning*, *information retrieval*, dan pengolahan citra karena mengukur tingkat kemiripan berdasarkan sudut yang terbentuk antara dua vektor, sehingga hasil perhitungan tidak dipengaruhi oleh besar kecilnya nilai vektor, melainkan oleh kesamaan arah kedua vektor. Semakin kecil sudut antara dua vektor, maka semakin tinggi tingkat kemiripannya [13].

Pada penerapannya, *cosine similarity* bekerja dengan menghitung nilai *cosinus* dari sudut antara dua vektor. Nilai hasil perhitungan berada pada rentang −1 hingga 1. Nilai 1 menunjukkan bahwa kedua vektor memiliki arah yang sama atau sangat mirip, nilai 0 menunjukkan bahwa kedua vektor tidak memiliki kemiripan, sedangkan nilai −1 menunjukkan bahwa kedua vektor memiliki arah yang berlawanan [14].




GAMBAR 4
Ilustrasi *cosine similarity*

Secara matematis, perhitungan *cosine similarity* dapat dinyatakan menggunakan persamaan (4) sebagai berikut.

$$\cos(\theta) = \frac{A \cdot B}{\| A \| \ \| B \|} \qquad (4)$$

dengan $A$ dan $B$ merupakan dua vektor yang akan dibandingkan, $A \cdot B$ adalah hasil *dot product* antara kedua vektor, sedangkan $\| A \|$ dan $\| B \|$ menyatakan panjang atau magnitudo masing-masing vektor [14].

E. Evaluasi Model

Evaluasi model merupakan tahapan yang dilakukan untuk mengukur kemampuan suatu model dalam menghasilkan prediksi yang sesuai dengan kondisi sebenarnya. [15].

Salah satu metode yang dinilai efektif digunakan dalam mengevaluasi model klasifikasi adalah *Confusion Matrix*. *Confusion Matrix* merupakan tabel yang membandingkan hasil prediksi model dengan kondisi sebenarnya (*ground truth*), sehingga mampu menunjukkan jumlah prediksi yang benar maupun salah untuk setiap kelas. Berdasarkan informasi tersebut, berbagai metrik evaluasi seperti *Accuracy*, *Precision*, *Recall*, dan *F1-score* dapat dihitung

sehingga performa model dapat dianalisis secara lebih menyeluruh [18]. Adapun bentuk umum *Confusion Matrix* ditunjukkan pada Gambar 5.




GAMBAR 5
Struktur *Confusion Matrix*

Berdasarkan Gambar 5. *Confusion Matrix* digunakan untuk menilai kinerja model klasifikasi dengan membandingkan hasil prediksi terhadap label sebenarnya atau *ground truth*. Metode ini memiliki empat komponen utama, yaitu *True Positive* (TP), *True Negative* (TN), *False Positive* (FP), dan *False Negative* (FN). TP menunjukkan data positif yang diprediksi dengan benar sebagai positif, sedangkan TN menunjukkan data negatif yang diprediksi dengan benar sebagai negatif. Sementara itu, FP merupakan data negatif yang keliru diprediksi sebagai positif, dan FN merupakan data positif yang keliru diprediksi sebagai negatif. Keempat komponen tersebut digunakan sebagai dasar untuk menghitung metrik *Accuracy, Precision, Recall*, dan *F1-score* dalam mengevaluasi performa model klasifikasi [19].

Metrik pertama yang umum digunakan adalah *Accuracy*, yaitu ukuran yang menunjukkan proporsi keseluruhan prediksi yang berhasil dilakukan dengan benar terhadap seluruh data uji. *Accuracy* memberikan gambaran umum mengenai tingkat ketepatan model, namun metrik ini dapat memberikan hasil yang bias apabila data yang digunakan memiliki distribusi kelas yang tidak seimbang (*imbalanced dataset*) [19]. Secara matematis, perhitungan *Accuracy* dinyatakan pada Persamaan (5).

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \quad (5)$$

Selanjutnya, *Precision* digunakan untuk mengukur tingkat ketepatan model dalam memberikan prediksi positif. Nilai *precision* menunjukkan seberapa besar proporsi data yang diprediksi positif benar-benar termasuk ke dalam kelas positif. *Precision yang tinggi menunjukkan bahwa model memiliki tingkat False Positive yang rendah* [17]. Secara matematis, perhitungan *Precision* dinyatakan pada Persamaan (6).

$$Precision = \frac{TP}{TP + FP} \quad (6)$$

Metrik berikutnya adalah *Recall*, yaitu ukuran yang menunjukkan kemampuan model dalam menemukan seluruh data yang benar-benar termasuk ke dalam kelas positif. *Recall* yang tinggi menunjukkan bahwa model mampu meminimalkan jumlah *False Negative* sehingga sebagian besar data positif berhasil dikenali [41]. Secara matematis, perhitungan *Recall* dinyatakan pada Persamaan (7).

$$Recall = \frac{TP}{TP + FN} \quad (7)$$

Untuk memperoleh penilaian yang mempertimbangkan keseimbangan antara *Precision* dan *Recall* digunakan *F1-score*. Nilai *F1-score* merupakan rata-rata harmonik dari kedua metrik tersebut sehingga banyak digunakan ketika distribusi data tidak seimbang. Semakin tinggi nilai *F1-score*, semakin baik keseimbangan antara kemampuan model dalam menghasilkan prediksi positif yang tepat dan kemampuan mendeteksi seluruh data positif [17]. Secara matematis, perhitungan *F1-score* dinyatakan pada Persamaan (8).

$$F1\text{-}Score = \frac{2 \times Precision \times Recall}{Precision + Recall} \quad (8)$$

Berdasarkan uraian tersebut, *Accuracy, Precision*, *Recall*, dan *F1-score* saling melengkapi dalam memberikan gambaran mengenai performa suatu model. *Accuracy* menunjukkan tingkat ketepatan prediksi secara keseluruhan, *Precision* mengukur ketepatan prediksi positif, *Recall* mengukur kemampuan menemukan seluruh data positif, sedangkan *F1-score* memberikan keseimbangan antara *Precision* dan *Recall*. Oleh karena itu, penggunaan keempat metrik tersebut secara bersamaan mampu memberikan evaluasi yang lebih komprehensif terhadap performa model klasifikasi maupun sistem *image retrieval* [16][17][18]

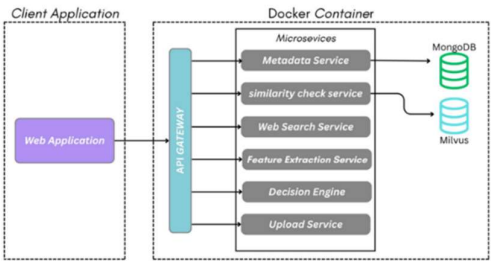
## III. METODE

### A. Desain Sistem

Sistem deteksi kemiripan karya citra digital berbasis *artificial intelligence* dirancang menggunakan beberapa komponen utama yang saling terintegrasi. Sistem dikembangkan dengan *backend* berbasis Python, penyimpanan data vektor menggunakan Milvus, pengelolaan metadata menggunakan MongoDB, serta penerapan metode *image processing* berbasis CNN dan CLIP

### B. Arsitektur Sistem

Arsitektur aplikasi yang dikembangkan dirancang menggunakan pendekatan *microservices* untuk meningkatkan modularitas, skalabilitas, dan kemudahan pemeliharaan sistem. Secara umum, arsitektur ini terdiri dari tiga komponen utama, yaitu *Client Application*, API *Gateway*, dan kumpulan layanan *microservices* yang dijalankan di dalam Docker *Container* serta terintegrasi dengan *database*. Gambaran keseluruhan arsitektur aplikasi dapat dilihat pada Gambar 6 berikut.



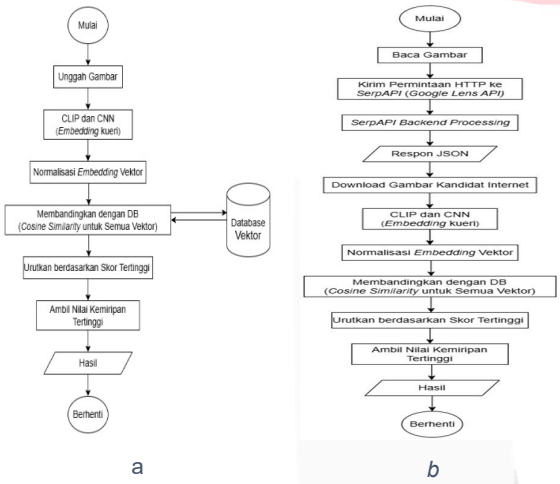

GAMBAR 6
Arsitektur sistem deteksi kemiripan citra

Di dalam lingkungan Docker *Container*, sistem dijalankan menggunakan arsitektur *microservices*, dimana setiap layanan memiliki fungsi yang spesifik dan saling terintegrasi. *Upload Service* bertanggung jawab untuk menerima data citra yang diunggah oleh pengguna. Selanjutnya, citra tersebut diproses oleh *Feature Extraction Service* untuk menghasilkan fitur citra menggunakan model *machine learning*. Hasil fitur disimpan dan dibandingkan

melalui Milvus sebagai database vektor yang digunakan dalam proses pencarian kemiripan. Proses pembandingan tersebut dilakukan oleh *Similarity Check Service* untuk memperoleh tingkat kemiripan antara citra yang diunggah dengan data citra yang telah tersimpan. *Web Search Service* digunakan untuk melakukan pencarian referensi citra dari sumber eksternal sebagai data pembanding tambahan

C. *Activity Diagram Similarity Detection* internal

Pada *similarity detection* internal, citra yang diunggah oleh pengguna terlebih dahulu diproses menggunakan model CLIP dan CNN untuk menghasilkan representasi fitur berupa vektor *embedding*. Vektor *embedding* yang diperoleh kemudian dinormalisasi dan dibandingkan dengan seluruh vektor citra yang tersimpan pada basis data vektor untuk memperoleh tingkat kemiripan antar citra. Selanjutnya, sistem mengurutkan hasil perbandingan berdasarkan nilai kemiripan tertinggi dan mengambil citra dengan skor tertinggi sebagai hasil pencarian. Nilai kemiripan tersebut kemudian digunakan sebagai dasar dalam penentuan status kemiripan citra sesuai dengan *threshold* yang telah ditetapkan oleh sistem. Alur proses *Similarity Detection* Internal ditunjukkan pada Gambar 7 (a) berikut.




GAMBAR 7
Diagram aktivitas deteksi kemiripan: internal (a) dan eksternal (b)

Alur proses *Similarity Detection* External ditunjukkan pada Gambar 7 (b). Pada *similarity* eksternal, citra yang diunggah oleh pengguna terlebih dahulu dibaca oleh sistem dan dikirim sebagai permintaan HTTP *request* ke SerpAPI yang memanfaatkan layanan Google Lens untuk melakukan pencarian citra serupa di internet. Hasil pencarian kemudian diterima dalam bentuk JSON *response* yang berisi informasi mengenai kandidat citra yang ditemukan pada sumber eksternal. Selanjutnya, sistem mengunduh citra kandidat tersebut dan memprosesnya menggunakan model CLIP dan CNN untuk menghasilkan vektor *embedding.* Vektor yang dihasilkan kemudian dinormalisasi dan dibandingkan dengan vektor citra kueri untuk memperoleh nilai kemiripan. Hasil perbandingan diurutkan berdasarkan skor kemiripan tertinggi, kemudian sistem mengambil nilai kemiripan tertinggi sebagai dasar penentuan hasil deteksi kemiripan eksternal yang selanjutnya ditampilkan kepada pengguna.

D. Metode pengukuran backend

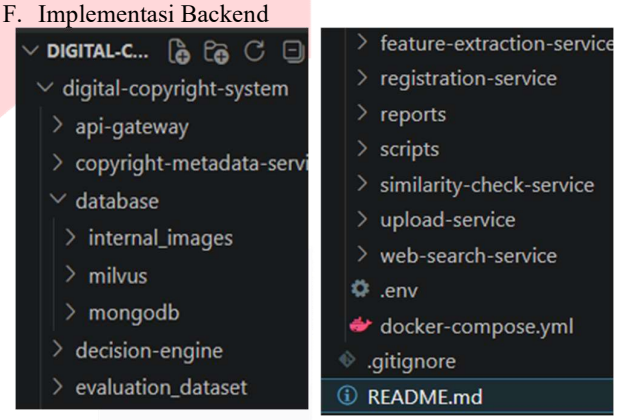
Pada pengukuran *backend*, dilakukan pengujian untuk memastikan bahwa layanan *backend* berjalan sesuai desain sistem, mencakup fungsionalitas API, alur komunikasi antar layanan, serta pengolahan data yang dilakukan oleh *backend.*

E. Metode pengukuran Image Processing

Pada bagian *image processing*, verifikasi difokuskan pada proses ekstraksi fitur dan penilaian kemiripan citra untuk memastikan mekanisme yang digunakan bekerja sesuai dengan spesifikasi sistem yaitu :

- Keberhasilan proses ekstraksi fitur menggunakan model CLIP dan CNN.
- Kesesuaian dimensi *embedding* yang dihasilkan dengan spesifikasi model.
- Ketepatan perhitungan nilai *similarity* yang terdiri atas CLIP *score*, CNN *score*, dan *final score*, serta
- Ketepatan sistem dalam menentukan status kemiripan berdasarkan *threshold* yang diterapkan.

F. Implementasi Backend




GAMBAR 8
Struktur implementasi *backend*

Backend pada sistem deteksi kemiripan karya citra digital digunakan untuk menangani proses utama sistem seperti pengelolaan data, komunikasi antar service, pengolahan request dari frontend, serta pengiriman response ke pengguna. Implementasi backend dilakukan untuk mendukung jalannya layanan sistem agar proses pengolahan data dapat berjalan secara terstruktur dan terintegrasi. Pada bagian ini dijelaskan implementasi struktur backend, endpoint API, validasi data, serta mekanisme *request* dan *response* pada sistem

G. Implementasi *Image Processing*

Implementasi proses *Image Processing* dilakukan untuk menghitung tingkat kemiripan gambar pada sistem. Tahapan ini meliputi *pre-processing* citra, *feature extraction*, pembentukan *embedding vector*, perhitungan *cosine similarity*, dan penentuan hasil berdasarkan *threshold similarity*. Seluruh proses tersebut digunakan agar sistem dapat menentukan apakah karya termasuk aman, perlu *review,* atau memiliki kemiripan tinggi.

H. Penentuan Bobot Perhitungan Final Score

Setelah proses ekstraksi fitur menggunakan model CLIP dan CNN selesai dilakukan, hasil *similarity* dari kedua model digabungkan untuk menghasilkan nilai *final score*. Pada

penelitian ini digunakan kombinasi bobot antara CNN dan CLIP untuk memperoleh keseimbangan antara kemiripan visual dan kemiripan semantik. Penentuan bobot dilakukan melalui proses *trial and error* menggunakan dataset pengujian internal dengan menguji beberapa kombinasi bobot CNN dan CLIP. Hasil pengujian kombinasi bobot ditunjukkan pada Tabel 1 berikut.

TABEL
Hasil pengujian bobot CNN dan CLIP

| Bobot CNN:CLIP | *Accuracy* (%) | *Precision* (%) | *Recall* (%) | *F1-score* (%) | TP | FP | FN | TN |
|---|---|---|---|---|---|---|---|---|
| 5:5 | 96,66 | 98,70 | 94,56 | 96,58 | 1513 | 20 | 87 | 1580 |
| 6:4 | 96,31 | 99,01 | 93,56 | 96,21 | 1497 | 15 | 103 | 1585 |
| 7:3 | 95,94 | 99,33 | 92,50 | 95,79 | 1480 | 10 | 120 | 1590 |
| 8:2 | 95,53 | 99,32 | 91,69 | 95,35 | 1467 | 10 | 133 | 1590 |
| 9:1 | 93,63 | 99,57 | 87,62 | 93,22 | 1402 | 6 | 198 | 1594 |

Berdasarkan Tabel 1, bobot 7:3 dipilih karena mampu mengurangi *false positive* sebesar 50%, dari 20 menjadi 10 kasus, dengan penurunan *accuracy* yang relatif kecil, yaitu 0,72%. Nilai *recall* juga tetap di atas 92%, sehingga sistem masih mampu mendeteksi citra serupa dengan baik. Konfigurasi ini memberikan keseimbangan antara CNN dalam mengenali kemiripan visual dan CLIP dalam mengenali kemiripan semantik.

## IV. HASIL DAN ANALISIS PENGUJIAN

### A. Hasil Pengujian *Backend*

Pengujian *backend* dilakukan untuk memastikan bahwa layanan b*ackend* mampu menerima *request*, melakukan validasi data masukan, meneruskan data ke layanan terkait, serta mengembalikan response sesuai dengan spesifikasi sistem. Pengujian dilakukan melalui beberapa endpoint API yang digunakan dalam proses deteksi kemiripan citra digital. Hasil pengujian *backend* disajikan pada Tabel 2 berikut.

TABEL 1
Hasil pengujian *endpoint backend*

| No | Skenario Pengujian | Langkah Pengujian | Ekspetasi Hasil | Hasil Aktual | Status |
|---|---|---|---|---|---|
| 1 | *Health Check Backend* | Mengakses *endpoint* `/health` atau *endpoint* pengecekan layanan | *Backend* mengembalikan status aktif atau berjalan | *Endpoint* berhasil diakses dan *backend* mengembalikan *status code* 200 dengan *response* yang menunjukkan bahwa layanan aktif. | Berhasil |
| 2 | *Upload* gambar valid | Mengunggah file gambar JPG, PNG, atau JPEG melalui *endpoint upload* | *Backend* menerima file dan memproses gambar | File gambar berhasil diterima oleh *backend*. Sistem memproses gambar dan mengembalikan *response* berupa data hasil pengecekan, seperti `check_id`, `similarity_result`, dan `decision_result`. | Berhasil |
| 3 | *Upload* file tidak valid | Mengunggah file yang tidak termasuk dalam format gambar | *Backend* menolak file dan menampilkan pesan kesalahan (*error message*) | File tidak valid berhasil ditolak oleh *backend*. Sistem mengembalikan *response error* yang menunjukkan bahwa format file tidak sesuai dengan ketentuan sistem. | Berhasil |
| 4 | Struktur *respons* API | Memeriksa *response JSON* setelah proses unggah gambar | *Response* berisi `check_id`, `similarity_result`, `decision_result`, dan data pendukung lainnya | *Response JSON* yang dikembalikan *backend* memiliki struktur data yang sesuai, yaitu memuat identitas pengecekan, hasil *similarity*, status keputusan, dan informasi pendukung lainnya. | Berhasil |
| 5 | Integrasi antar layanan | Mengirim gambar hingga hasil *similarity* ditampilkan | *Backend* berhasil meneruskan data ke modul *image processing* dan *database* | *Backend* berhasil meneruskan gambar ke modul *image processing* untuk proses ekstraksi fitur dan perhitungan *similarity*. Hasil pengolahan kemudian dapat dikembalikan dan digunakan oleh sistem. | Berhasil |
| 6 | Waktu respons *backend* | Mencatat waktu sejak *request* dikirim hingga | Waktu respons sistem dapat tercatat dalam satuan detik | Waktu respons *backend* berhasil dicatat dari proses *request* hingga | Berhasil |

Penelitian ini juga menguji waktu proses *endpoint /upload* berdasarkan jumlah data referensi dalam basis data, yaitu 25, 50, 100, 150, dan 200 data. Pengujian dilakukan untuk mengetahui pengaruh pertambahan data terhadap waktu deteksi kemiripan citra. *Endpoint /upload* dipilih karena mencakup seluruh proses utama, mulai dari unggah dan *pre-processing* citra, ekstraksi fitur CNN dan CLIP, pembentukan *embedding*, pencarian pada *vector database*, hingga perhitungan *final score*. Karena memiliki beban komputasi lebih besar dibandingkan endpoint CRUD metadata, *endpoint* ini dinilai paling mewakili performa sistem secara keseluruhan. Hasil pengujian ditunjukkan pada Tabel 3.

TABEL 3
Waktu *respons endpoint /upload* berdasarkan jumlah data referensi

| No | Jumlah Data | *Endpoint* | *Status Code* | *Response Time (second)* |
|---|---|---|---|---|
| 1 | 25 | `/upload` | 200 | 9,29 |
| 2 | 50 | `/upload` | 200 | 9,871 |
| 3 | 100 | `/upload` | 200 | 10,587 |
| 4 | 150 | `/upload` | 200 | 7,04 |
| 5 | 200 | `/upload` | 200 | 7,305 |

### B. Hasil Pengujian Image Processing

Evaluasi kinerja deteksi kemiripan citra dilakukan untuk mengetahui kemampuan sistem dalam menentukan tingkat kemiripan antar citra berdasarkan nilai *clip_score, cnn_score*, dan *final_score* yang dihasilkan. Nilai tersebut digunakan untuk mengklasifikasikan citra ke dalam kategori *High Similarity, Medium Similarity*, atau *Low Similarity* sesuai dengan skenario pengujian yang telah ditetapkan. Rekapitulasi hasil evaluasi deteksi kemiripan citra ditunjukkan pada Tabel 4.

TABEL 4
Hasil pengujian deteksi kemiripan citra

| No | Skenario | Kondisi Data Uji | clip_score | cnn_score | final_score | Prediksi Sistem |
|---|---|---|---|---|---|---|
| 1 | Identik | Sama dengan data internal | 100% | 100% | 100% | *High Similarity* |
| 2 | Identik | Sama dengan data internal | 100% | 100% | 100% | *High Similarity* |
| 3 | Identik | Sama dengan data internal | 100% | 100% | 100% | *High Similarity* |
| 4 | *Resize* | Ukuran gambar diubah | 99,99% | 99,99% | 99,99% | *High Similarity* |
| 5 | *Resize* | Resolusi gambar diubah | 100% | 100% | 100% | *High Similarity* |
| 6 | *Resize* | Gambar dikompresi | 99,99% | 100% | 99,99% | *High Similarity* |
| 7 | *Crop* | Sebagian area gambar dipotong | 97,39% | 97,25% | 97,39% | *High Similarity* |
| 8 | *Crop* | Objek utama masih terlihat | 94,33% | 90% | 91,73% | *High Similarity* |
| 9 | *Crop* | Area gambar dipotong lebih besar | 92,75% | 83,56% | 87,24% | *High Similarity* |
| 10 | Rotasi 90° | Gambar diputar 90 derajat | 87,87% | 88,77% | 88,41% | *High Similarity* |
| 11 | Rotasi 180° | Gambar diputar 180 derajat | 90,36% | 88,27% | 89,11% | *High Similarity* |
| 12 | Rotasi | Gambar diputar dengan sudut tertentu | 96,76% | 87,78% | 91,37% | *High Similarity* |
| 13 | Satu kategori berbeda | Kategori sama, desain berbeda | 82,3% | 56,48% | 66,81% | *Low Similarity* |
| 14 | Satu kategori berbeda | Kategori sama, objek berbeda | 80,94% | 65,6% | 71,73% | *Medium Similarity* |
| 15 | Satu kategori berbeda | Tema mirip, desain berbeda | 89,71% | 44,89% | 62,82% | *Low Similarity* |
| 16 | Beda Kategori | Tidak berkaitan dengan data internal | 80,82% | 55,15% | 65,42% | *Low Similarity* |
| 17 | Beda Kategori | Visual tidak memiliki hubungan | 0% | 42,17% | 25,3% | *Low Similarity* |
| 18 | *File* tidak valid | File PDF | - | - | - | *File* ditolak sistem |

Selain mengevaluasi hasil deteksi kemiripan citra, pengujian juga dilakukan terhadap waktu pemrosesan pada setiap skenario data uji. Pengukuran ini bertujuan untuk mengetahui performa sistem dalam memproses citra dan menghasilkan hasil deteksi kemiripan pada berbagai kondisi pengujian. Waktu proses dihitung sejak citra diunggah hingga sistem menampilkan hasil deteksi kemiripan. Rekapitulasi waktu pemrosesan pada setiap data uji ditunjukkan pada Tabel 5.

TABEL 5
Waktu pemrosesan setiap skenario pengujian

| No | Skenario | Kondisi Data Uji | Waktu Proses |
|---|---|---|---|
| 1 | Identik | Sama dengan data internal | 5,31 detik |
| 2 | Identik | Sama dengan data internal | 18,15 detik |
| 3 | Identik | Sama dengan data internal | 8,38 detik |
| 4 | *Resize* | Ukuran gambar diubah | 8,23 detik |
| 5 | *Resize* | Resolusi gambar diubah | 5,20 detik |
| 6 | *Resize* | Gambar dikompresi | 6,39 detik |
| 7 | *Crop* | Sebagian area gambar dipotong | 13,86 detik |
| 8 | *Crop* | Objek utama masih terlihat | 9,49 detik |
| 9 | *Crop* | Area gambar dipotong lebih besar | 7,49 detik |
| 10 | Rotasi | Gambar diputar 90 derajat | 13,70 detik |
| 11 | Rotasi | Gambar diputar 180 derajat | 10,02 detik |
| 12 | Rotasi | Gambar diputar dengan sudut tertentu | 15,16 detik |
| 13 | Satu kategori berbeda | Kategori sama, desain berbeda | 4,31 detik |
| 14 | Satu kategori berbeda | Kategori sama, objek berbeda | 7,81 detik |
| 15 | Satu kategori berbeda | Tema mirip, desain berbeda | 5,61 detik |
| 16 | Beda Kategori | Tidak berkaitan dengan data internal | 8,69 detik |
| 17 | Beda Kategori | Visual tidak memiliki hubungan | 10,77 detik |
| 18 | *File* tidak valid | File PDF | - |

Selain pengujian pada Tabel 5 dan Tabel 6, keandalan sistem juga dievaluasi menggunakan dataset eksternal berisi 3.200 pasangan citra. Dataset berasal dari 200 citra *The Metropolitan Museum of Art* dan *Rijksmuseum* yang diberi delapan transformasi, yaitu *blur, brightness, compress, contrast, crop, resize, rotate*, dan *watermark*. Setiap transformasi terdiri atas 200 pasangan *plagiarized* dan 200 pasangan *not plagiarized*. Hasil confusion matrix keseluruhan ditunjukkan pada Tabel 6.

TABEL 6
Confusion Matrix keseluruhan

| *Actual \ predicted* | *Plagiarized* | *Not Plagiarized* |
|---|---|---|
| *Plagiarized* | 1480 | 120 |
| *Not Plagiarized* | 10 | 1590 |

Berdasarkan *confusion matrix* keseluruhan, dilakukan perhitungan metrik evaluasi yang meliputi *accuracy*, *precision, recall, dan F1-score* untuk mengetahui kehandalan sistem dalam mendeteksi kemiripan citra. Hasil perhitungan evaluasi ditunjukkan pada Tabel 7 berikut.

TABEL 7
Hasil evaluasi keandalan sistem secara keseluruhan

| No | Metrik Evaluasi | Nilai |
|---|---|---|
| 1 | *Accuracy* | 95,94% |
| 2 | *Precision* | 99,33% |
| 3 | *Recall* | 92,50% |
| 4 | *F1-score* | 85,79% |

Selanjutnya dilakukan juga evaluasi pada masing-masing jenis transformasi untuk mengetahui performa sistem pada setiap kondisi perubahan visual. Rekapitulasi nilai *true positive* (TP), *false positive* (FP), *true negative* (TN) dan *false negative* (FN) pada setiap transformasi ditunjukkan pada Tabel 8 berikut.

TABEL 8
*Confusion Matrix* per transformasi

| Transformasi | TP | FP | FN | TN |
|---|---|---|---|---|
| *Blur* | 191 | 1 | 9 | 199 |
| *Brightness* | 199 | 2 | 1 | 198 |
| *Compress* | 200 | 1 | 0 | 199 |
| *Contrast* | 200 | 3 | 0 | 197 |
| *Crop* | 193 | 1 | 7 | 199 |
| *Resize* | 200 | 2 | 0 | 198 |
| *Rotate* | 178 | 0 | 22 | 200 |
| *Watermark* | 199 | 0 | 81 | 200 |

Berdasarkan nilai *confusion matrix* pada masing-masing transformasi, dilakukan perhitungan *accuracy*, *precision*, *recall*, dan *F1-score* untuk setiap jenis transformasi. Hasil perhitungan evaluasi ditunjukkan pada Tabel 9 berikut.

TABEL 9
Hasil evaluasi keandalan sistem per transformasi

| Transformasi | *Accuracy* | *Precision* | *Recall* | *F1-score* |
|---|---|---|---|---|
| *Blur* | 97,50% | 99,48% | 95,50% | 97,45% |
| *Brightness* | 99,25% | 99% | 99,50% | 99,25% |
| *Compress* | 99,75% | 99,50% | 100% | 99,75% |
| *Contrast* | 99,25% | 98,52% | 100% | 99,26% |
| *Crop* | 98% | 99,48% | 96,50% | 97,97% |
| *Resize* | 99,50% | 99,01% | 100% | 99,50% |
| *Rotate* | 94,50% | 100% | 89% | 94,18% |
| *Watermark* | 79,75% | 100% | 59,50% | 74,61% |

C. Analisis Hasil Pengujian Backend

Berdasarkan hasil pengujian, seluruh fungsi backend berjalan sesuai dengan spesifikasi sistem. Backend mampu menerima dan memvalidasi *request*, mengembalikan *response* yang sesuai, serta terintegrasi dengan modul *image processing* dan *database*. Sistem juga dapat menangani kesalahan, seperti format *file* yang tidak sesuai, dengan memberikan pesan yang informatif sehingga data tidak valid tidak diproses lebih lanjut. Pengujian waktu respons menunjukkan bahwa *backend* memerlukan waktu sekitar 8,81 detik untuk memproses satu permintaan hingga menghasilkan hasil deteksi kemiripan. Waktu tersebut dipengaruhi oleh proses ekstraksi fitur menggunakan model

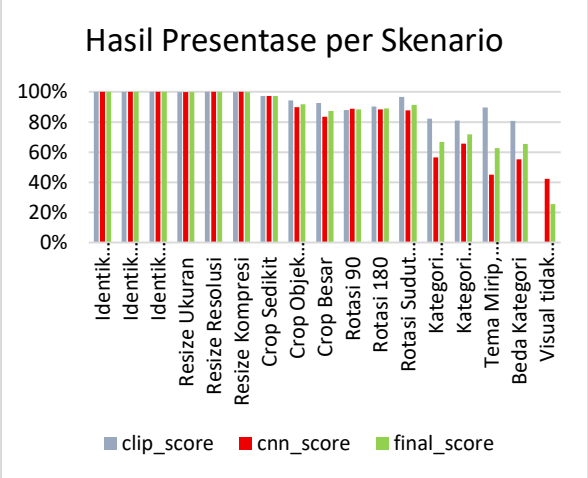
CNN dan CLIP, pencarian pada *vector database*, serta proses pengambilan keputusan berdasarkan nilai *similarity*. Secara keseluruhan, hasil pengujian menunjukkan bahwa layanan backend telah berjalan sesuai dengan kebutuhan sistem.

D. Analisis Hasil Pengujian Image processing

Berdasarkan hasil pengujian, sistem berhasil melakukan ekstraksi fitur citra menggunakan model CLIP dan CNN. Kedua model tersebut menghasilkan representasi fitur yang digunakan dalam proses perhitungan *similarity score*. Sistem juga berhasil menghasilkan nilai *clip_score, cnn_score*, dan *final_score* yang digunakan sebagai dasar dalam penentuan status kemiripan citra. Hasil pengujian menunjukkan bahwa integrasi antara model CLIP, CNN, dan metode perhitungan *similarity* telah berjalan dengan baik dan sesuai dengan rancangan sistem. Selain itu, sistem juga berhasil menentukan status kemiripan citra berdasarkan nilai threshold yang telah ditetapkan, sehingga proses klasifikasi citra dapat dilakukan secara otomatis.

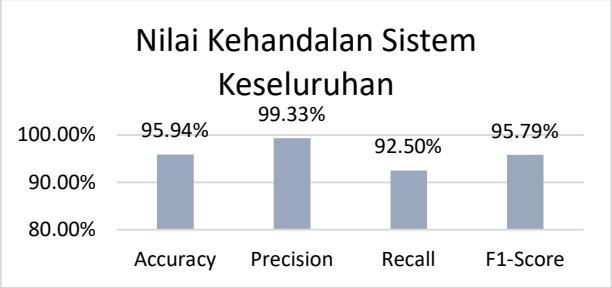
Berdasarkan hasil pengujian, sistem mampu mendeteksi citra identik dengan nilai *final_score* sebesar 100%. Pada skenario *resize, crop*, dan rotasi, sistem masih menghasilkan nilai *final_score* yang tinggi, yaitu di atas 87%. Hal ini menunjukkan bahwa sistem memiliki kemampuan yang baik dalam mengenali perubahan minor pada citra tanpa kehilangan karakteristik utama dari gambar.

Sebaliknya, pada skenario satu kategori berbeda dan beda kategori, nilai *final_score* mengalami penurunan yang cukup signifikan. Hasil tersebut menunjukkan bahwa sistem mampu membedakan citra yang memiliki kemiripan visual dengan citra yang tidak memiliki hubungan secara visual. Selain itu, sistem juga berhasil menolak *file* yang tidak sesuai format sehingga mencegah proses pengolahan terhadap data yang tidak valid. Perbandingan nilai *clip_score, cnn_score*, dan *final_score* pada setiap skenario pengujian ditunjukkan pada Gambar 9.



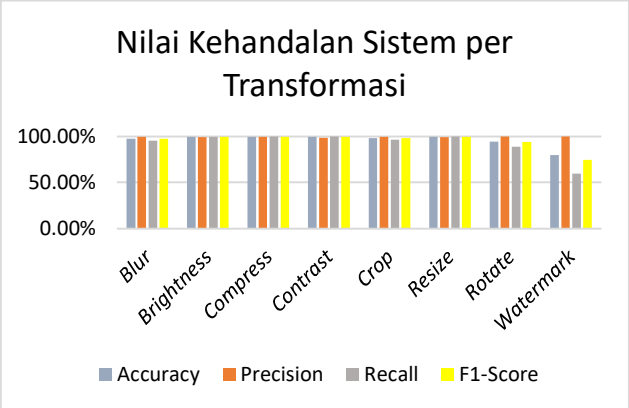

GAMBAR 9
Perbandingan *clip_score, cnn_score*, dan *final_score* pada setiap skenario

Berdasarkan hasil evaluasi kehandalan sistem secara keseluruhan, sistem memperoleh nilai *accuracy* sebesar 95,94%, *precision* sebesar 99,33%, *recall* sebesar 92,50%, dan *F1-score* sebesar 95,79%. Secara keseluruhan, hasil tersebut menunjukkan bahwa sistem memiliki tingkat kehandalan yang baik dalam mendeteksi kemiripan citra pada berbagai kondisi pengujian. Nilai kehandalan sistem keseluruhan ditunjukkan pada Gambar 10 berikut.




GAMBAR 10
Nilai keandalan sistem secara keseluruhan

Berdasarkan hasil evaluasi pada setiap jenis transformasi, sistem menunjukkan performa yang sangat baik pada transformasi *brightness, compress, contrast, resize, crop*, dan *blur,* dengan nilai a*ccuracy* di atas 97%. Hasil tersebut menunjukkan bahwa perubahan ukuran, kompresi, tingkat pencahayaan, kontras, pemotongan, maupun efek blur tidak memberikan pengaruh yang signifikan terhadap kemampuan sistem dalam mengenali kemiripan citra. Sementara itu, transformasi *rotate* menunjukkan penurunan performa, terutama pada nilai *recall* yang mencapai 89,00%, yang mengindikasikan bahwa masih terdapat beberapa pasangan citra *plagiarized* yang tidak berhasil dikenali setelah mengalami rotasi. Performa terendah diperoleh pada transformasi *watermark*, dengan nilai *accuracy* sebesar 79,75%, *recall* sebesar 59,50%, dan *F1-score* sebesar 74,61%. Hal tersebut menunjukkan bahwa penambahan watermark memberikan pengaruh yang lebih besar terhadap proses ekstraksi fitur sehingga menurunkan kemampuan sistem dalam mendeteksi kemiripan citra dibandingkan transformasi lainnya. Nilai kehandalan sistem per transformasi ditunjukkan pada Gambar 11 berikut.




GAMBAR 11
Nilai keandalan sistem per transformasi

## IV. KESIMPULAN

Berdasarkan hasil perancangan, implementasi, dan pengujian yang telah dilakukan, maka dapat disimpulkan bahwa. Sistem deteksi kemiripan karya citra digital untuk seleksi pendaftaran hak cipta berbasis *Artificial Intelligence* berhasil dikembangkan dalam bentuk aplikasi berbasis web menggunakan model *Contrastive Language-Image Pre-training* (CLIP) dan *Convolutional Neural Network* (CNN). Sistem juga berhasil diintegrasikan dengan MongoDB

sebagai penyimpanan metadata dan Milvus sebagai basis data vektor untuk menyimpan embedding citra. Sistem mampu melakukan deteksi kemiripan citra berdasarkan nilai *clip_score, cnn_score,* dan *final_score* serta mengklasifikasikan hasil deteksi ke dalam kategori *High Similarity, Medium Similarity*, dan *Low Similarity*. Hasil pengujian menunjukkan bahwa sistem mampu mengenali citra identik serta citra yang mengalami perubahan seperti *resize, crop,* dan rotasi dengan baik. Berdasarkan hasil evaluasi menggunakan data internal Bandung Techno Park (BTP), sistem memperoleh nilai *accuracy* sebesar 95,94%, *precision* sebesar 99,33%, *recall* sebesar 92,50%, dan *F1-score* sebesar 95,79%. Hasil tersebut menunjukkan bahwa sistem memiliki kemampuan yang baik dalam mendeteksi kemiripan citra digital.

## REFERENSI


[1] M. Yanu, B. Yuwono, dan D. Boedi, "Dasar Pengolahan Digital Edisi 2022," Kedua., UPN Veteran Yogyakarta, 2022.

[2] D. T. Wijayanty *dkk.*, "Aspek Hukum dalam Perlindungan Hak Cipta di Era Digital," *Mediasi*, vol. 4, no. 2, hlm. 027–033, Feb 2025, [Daring]. Tersedia pada: http://jurnalilmiah.org/journal/index.php/mediasi

[3] S. Mondal, P. De, S. Malakar, dan R. Sarkar, "OMRNet: A lightweight deep learning model for optical mark recognition," *Multimed. Tools Appl.*, vol. 83, no. 5, hlm. 14011–14045, Feb 2024, doi: 10.1007/s11042-023-15408-8.

[4] A. Sukardi, D. Trisnawarman, dan N. J. Perdana, "Perancangan Sistem Informasi Pendaftaran Hak Cipta," *Jutisi*, vol. 13, hlm. 2348–2359, Des 2024.

[5] X. Liu, J. Liang, Z. Y. Wang, Y. Te Tsai, C. C. Lin, dan C. C. Chen, "Content-based image copy detection using convolutional neural network," *Electronics (Switzerland)*, vol. 9, no. 12, hlm. 1–16, Des 2020, doi: 10.3390/electronics9122029.

[6] W. Wang, Y. Sun, dan Y. Yang, "A Benchmark and Asymmetrical-Similarity Learning for Practical Image Copy Detection," Des 2022, [Daring]. Tersedia pada: http://arxiv.org/abs/2205.12358

[7] Z. Li, F. Liu, W. Yang, S. Peng, dan J. Zhou, "A Survey of Convolutional Neural Networks: Analysis, Applications, and Prospects," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 33, no. 12, hlm. 6999–7019, Des 2022, doi: 10.1109/TNNLS.2021.3084827.

[8] A. Hasan, T. Bhargav, V. Sandeep, V. S. Reddy, dan R. Ajay, "IMAGE CLASSIFICATION USING CONVOLUTIONAL NEURAL NETWORKS." [Daring]. Tersedia pada: http://www.ijmert.com

[9] Purwono, A. Ma'arif, W. Rahmaniar, H. I. K. Fathurrahman, A. Z. K. Frisky, dan Q. M. U. Haq, "Understanding of Convolutional Neural Network (CNN): A Review," *International Journal of Robotics and Control Systems*, vol. 2, no. 4, hlm. 739–748, 2022, doi: 10.31763/ijrcs.v2i4.888.

[10] C. A. Sari, L. Hamidah, S. Pradana, dan E. H. Rachmawanto, "Identifikasi Citra Jenis Rempah-Rempah Menggunakan Arsitektur RestNet50," *LogicLink: Journal of Artificial Intelligence and Multimedia in Informatics*, vol. 2, no. 1, hlm. 27–41, 2025.

[11] A. Muriagista dan D. Kurniadi, "IMPLEMENTASI ARSITEKTUR RESNET50 PADA KLASIFIKASI MOTIF BATIK INDONESIA MENGGUNAKAN METODE CONVOLUTIONAL NEURAL NETWORK (CNN)," *Jurnal Rekayasa Sistem Informasi dan Teknologi*, vol. 2, no. 4, Mei 2025.

[12] "resnet50 — Torchvision main documentation." Diakses: 13 Juli 2026. [Daring]. Tersedia pada: https://docs.pytorch.org/vision/main/models/generated/torchvision.models.resnet50

[13] A. Radford *dkk.*, "Learning Transferable Visual Models From Natural Language Supervision," 2021. [Daring]. Tersedia pada: https://github.com/OpenAI/CLIP.

[14] T. Taipalus, "Vector database management systems: Fundamental concepts, use-cases, and current challenges," *Cogn. Syst. Res.*, vol. 85, Jun 2024, doi: 10.1016/j.cogsys.2024.101216.

[15] H. Steck, C. Ekanadham, dan N. Kallus, "Is Cosine-Similarity of Embeddings Really About Similarity?," dalam *WWW 2024 Companion - Companion Proceedings of the ACM Web Conference*, Association for Computing Machinery, Inc, Mei 2024, hlm. 887–890. doi: 10.1145/3589335.3651526.

[16] V. Kazakovtsev *dkk.*, "Fast Adaptive Approximate Nearest Neighbor Search with Cluster-Shaped Indices," *Big Data and Cognitive Computing*, vol. 9, no. 10, Okt 2025, doi: 10.3390/bdcc9100254.

[17] C. Miller, T. Portlock, D. M. Nyaga, dan J. M. O'Sullivan, "A review of model evaluation metrics for machine learning in genetics and genomics," 2024, *Frontiers Media SA*. doi: 10.3389/fbinf.2024.1457619.

[18] O. Rashid Khan dan J. I. Bhat, "Delving into the Depths of Image Retrieval Systems in the Light of Deep Learning: A Review," *Indian J. Sci. Technol.*, 2023, doi: 10.17485/IJST/v16i34.1341.

[19] R. Vishraj, S. Gupta, dan S. Singh, "A comprehensive review of content-based image retrieval systems using deep learning and hand-crafted features in medical imaging: Research challenges and future directions," *Computers and Electrical Engineering*, vol. 104, Des 2022, doi: 10.1016/j.compeleceng.2022.108450.